# Ayuda AppleScript

Contenido esencial

## Acerca de AppleScript

A medida que utilice su Mac, comprobará que realiza determinadas tareas repetidamente. Con AppleScript podrá crear scripts y aplicaciones para automatizar esas tareas repetitivas.

AppleScript es un potente y versátil lenguaje de programación que está integrado en OS X. Puede utilizarlo para crear funciones rápidas, automatizar tareas repetitivas o incluso diseñar aplicaciones personalizadas que le ahorren mucho tiempo. AppleScript le permite controlar las aplicaciones que admiten el uso de scripts.

Un script de AppleScript consiste en una serie de comandos en inglés que realizan acciones sobre objetos. Ejemplos de acciones: open, close, get the name of (es decir, abrir, cerrar, obtener el nombre de). Los objetos pueden ser, por ejemplo, una aplicación, una ventana o un ajuste de preferencias.

Un script puede controlar una o más aplicaciones. Para que una aplicación permita el uso de scripts, debe definir los términos que se pueden emplear en los scripts. El sistema OS X incluye varias aplicaciones que aceptan scripts.

Los scripts también pueden controlar aplicaciones a través de una red, de modo que permiten automatizar tareas en más de un ordenador.

De hecho, usted ya ha utilizado AppleScript sin saberlo. Muchas funciones del sistema y de las aplicaciones son, en realidad, scripts de AppleScript que se ejecutan dentro de una aplicación. Por ejemplo, en este sistema de ayuda, cada vez que hace clic sobre un enlace que abre una aplicación o un archivo, la acción la realiza un script de AppleScript.

**TEMAS RELACIONADOS**

Funcionamiento de AppleScript
Acerca de Editor AppleScript
Lenguaje AppleScript
Recursos sobre AppleScript en Internet
Entorno de desarrollo AppleScript Studio

---

Contenido esencial

## Acerca de Editor AppleScript

Editor AppleScript es una aplicación para crear scripts muy fácil de utilizar que permite leer, escribir, grabar y guardar scripts de AppleScript.

Estas son algunas características de Editor AppleScript:

- Ofrece una interfaz unificada y fácil de usar.
- Le permite buscar y reemplazar en el texto del script.
- Realiza un ajuste automático de las sentencias de gran longitud.
- Le permite deshacer varias acciones.

- Ofrece scripts de menú de función rápida para agilizar la redacción de scripts.
- Ofrece soporte para los servicios de las aplicaciones.
- Le permite acceder a scripts de aplicación mediante la paleta Biblioteca
- Ofrece un Asistente para Scripts que le ayuda a completar las sentencias.
- Ofrece el historial de resultados y el historial de registro de eventos para que pueda examinar qué ocurre en un script

**TEMAS RELACIONADOS**

Lenguaje AppleScript
Entorno de desarrollo AppleScript Studio

---

**Contenido esencial**

## Crear scripts de AppleScript

No necesita saber nada de programación para beneficiarse de AppleScript. De hecho, no necesita escribir ni una sola palabra de un script. El sistema OS X incluye cientos de prácticos scripts de AppleScript que puede comenzar a utilizar, y en Internet encontrará muchos más.

Si desea profundizar un poco más, puede adaptar los scripts existentes a sus necesidades con pequeñas modificaciones.

Por supuesto, obtendrá las máximas ventajas si crea sus propios scripts. Aprender a escribir scripts le permitirá crear herramientas útiles y automatizar tareas que le ahorrarán mucho tiempo y que harán más productivo su trabajo.

Puede modificar o crear sus propios scripts con Editor AppleScript, que proporciona un entorno sencillo donde escribir sus scripts. Todos los ejemplos de la Ayuda AppleScript se han realizado utilizando Editor AppleScript.

---

- Abrir Editor AppleScript.

  Escriba el texto siguiente en la ventana de Editor AppleScript.

  ```
  say "Welcome to AppleScript!"
  ```

  Haga clic en Ejecutar.

---

**TEMAS RELACIONADOS**

Acerca de Editor AppleScript
Entorno de desarrollo AppleScript Studio

---

**Contenido esencial**

## Funcionamiento de AppleScript

AppleScript funciona enviando mensajes, llamados "eventos Apple". Cuando se ejecuta un script, AppleScript envía eventos Apple al sistema o a las aplicaciones que llevan a cabo los comandos.

En muchos casos, las aplicaciones o el sistema devolverán un resultado a AppleScript. Por ejemplo, si un script envía una petición a una aplicación para obtener el nombre de un objeto, la aplicación devuelve el nombre. Si este comando no puede ejecutarse, la aplicación o AppleScript pueden devolver un mensaje de error. Puede ver los resultados en Editor

AppleScript.

En general, no es necesario tener grandes conocimientos acerca de la mecánica de los eventos Apple, ya que AppleScript los gestiona siempre en segundo plano. Usted solo necesita escribir y ejecutar sus scripts.

**TEMAS RELACIONADOS**

Acerca de AppleScript
Lenguaje AppleScript

---

**Contenido esencial**

## Applets

Un script que se guarda como aplicación se llama "aplicación AppleScript" o "applet". Los applets se ejecutan automáticamente al abrirse y realizan los comandos del script. Los applets pueden ser pequeñas funciones que contengan solo un par de comandos, o aplicaciones complejas que contengan miles de líneas.

Los applets pueden guardarse como editables o como solo ejecutables. Un applet editable sigue conteniendo el texto del script, por lo que puede editarse y volver a compilarse en cualquier momento. Para modificar un applet, arrástrelo hasta el icono de Editor AppleScript. Cuando se guarda un script como solo ejecutable, la parte de texto editable se elimina y el archivo ya no puede modificarse ni volver a compilarse.

Para modificar un applet que se ha guardado como editable, ábralo en Editor AppleScript o en otra aplicación de edición de AppleScript. Los applets requieren un gestor "on run" que inicie la ejecución del script cuando se abra el applet.

- En este ejemplo se muestra un gestor “on run”.

```
on run
        --first command
        --second command
end run
```

**TEMAS RELACIONADOS**

Droplets
Tipos de archivo de AppleScript

---

**Contenido esencial**

## Scripts compilados

Un script compilado es un archivo de AppleScript en un formato editable que se puede leer y modificar en Editor AppleScript o en otro programa de edición de AppleScript.

Al abrir un script compilado, el script se mostrará en Editor AppleScript, pero no se ejecutará automáticamente ninguno de los comandos. De este modo, se pueden realizar cambios en el script sin tener que ejecutarlo. A continuación, puede compilar el script, acción que lee y convierte todo el texto del archivo en código de AppleScript. Una vez compilado, el script se puede ejecutar. Un script compilado tiene la extensión ".scpt".

Cuando se guarda el archivo como script compilado, el texto se guarda junto con el código. De esa manera puede editar de nuevo el script compilado en cualquier momento.

**TEMAS RELACIONADOS**

Applets
Droplets
Tipos de archivo de AppleScript

---

Contenido esencial

## Droplets

Un droplet es un script de AppleScript que se ejecuta cuando se arrastran archivos o carpetas sobre su icono. El script realiza acciones sobre dichos archivos o carpetas.

Al igual que los applets, un droplet puede guardarse como editable o como solo ejecutable. Un droplet editable sigue conteniendo el texto del script, por lo que puede editarse y volver a compilarse en cualquier momento. Para modificar un droplet, arrástrelo hasta el icono del Editor de Scripts. Cuando se guarda un script como solo ejecutable, la parte de texto editable se elimina y el archivo ya no puede modificarse ni volver a compilarse.

Los droplets requieren un gestor "on open" que inicie la ejecución del script cuando se abra el droplet. El gestor "on open" posee un parámetro que contiene una lista de los elementos arrastrados sobre el droplet. Una parte del script del droplet examina la lista y otra parte actúa sobre los elementos. A menos que el droplet se haya diseñado para actuar sobre un solo elemento, incluirá un bucle de repetición que pasará por todos los elementos de la lista realizando las acciones en cada uno de ellos.

- A continuación se incluye un ejemplo de droplet con el gestor “on open” y una lista de parámetros.

```
on open (theItemsDropped)
        --first command
        --second command
end open
```

**TEMAS RELACIONADOS**

Applets
Scripts compilados
Tipos de archivo de AppleScript

---

Contenido esencial

## Usar diccionarios de AppleScript

Para saber qué comandos y objetos puede usar para controlar una aplicación, puede consultar el diccionario de dicha aplicación. Un diccionario contiene un conjunto de términos que una determinada aplicación es capaz de entender.

Puede ver el diccionario de una aplicación en Editor AppleScript seleccionando la aplicación en la ventana Biblioteca o arrastrando el icono de la aplicación al icono de Editor AppleScript.

Los términos de un diccionario están organizados en grupos de términos relacionados denominados "suites" o "conjuntos de instrucciones". Los comandos y los objetos están en texto en negrita.

1. En Editor AppleScript, seleccione Ventana > Biblioteca.

2. Haga doble clic en el nombre de la aplicación en la ventana Biblioteca.
3. Seleccione un conjunto de instrucciones (suite).

   Se visualizarán los términos incluidos y sus descripciones.

---

El lenguaje AppleScript está estructurado en diferentes categorías:

- Conjuntos: son grupos de términos relacionados. En el Diccionario de Editor AppleScript, los conjuntos se muestran en la primera columna y se designan mediante una letra naranja "S".
- Comandos: sirven para solicitar la ejecución de una acción o un resultado. Los comandos, que también se denominan "eventos", se muestran en la segunda columna, que se designa mediante una letra azul "C".
- Clases: son objetos que permiten crear scripts. Las descripciones de las clases contienen las propiedades y los elementos que integran una clase. Las clases aparecen en la segunda columna y se designan mediante una letra "C" de color lila.
- Propiedades: son objetos que contienen valores. Las propiedades aparecen en la tercera columna y se designan mediante una letra "P" de color lila.
- Elementos: son objetos contenidos dentro de otros objetos. Por ejemplo, un párrafo puede contener diversas palabras. Los elementos pueden aparecer en la tercera columna y se designan mediante una letra "E" de color naranja.

**TEMAS RELACIONADOS**

Definiciones de clase
Definiciones de comando
Utilizar AppleScript para enviar comandos a aplicaciones

**ABRIR**

Guía AppleScript Language Guide: referencia de comandos
Guía AppleScript Language Guide: referencia de clase

---

**Contenido esencial**

## Solucionar problemas

Si tiene problemas con el uso de AppleScript, explore los siguientes temas en busca de información para solucionarlos.

Si ocurre un error cuando un usuario hace clic en Cancelar en un cuadro de diálogo de selección de una lista (choose from list)

Si la acción de carpeta se ejecuta de manera inesperada

Si el script no reconoce un disco

---

**Comandos básicos**

## Definiciones de clase

Una definición de clase describe las propiedades y los elementos de los objetos que pueden utilizarse en los scripts. Una

clase puede compararse a un sustantivo de una frase, y la definición de clase es como la definición de dicho sustantivo en el diccionario.

Normalmente, en un diccionario de AppleScript una definición de una clase empieza por un término, en negrita, y continúa con una breve descripción de ese término en la misma línea.

Después puede verse una lista de elementos y objetos que pertenecen al objeto o que constituyen el mismo. No todas las definiciones de clase poseen lista de elementos porque algunos objetos de script contienen otros objetos de script y otros no.

La definición de la clase termina con una lista de propiedades o cualidades del objeto de script y sus valores.

Una entrada de una propiedad comienza por el nombre de la propiedad y va seguido de:

- Una descripción del formato de datos en el que se guarda el valor de la propiedad; por ejemplo: integer (número entero), point (una lista de dos números), string (datos de texto) y Boolean (verdadero o falso)
- Un indicador de tipo opcional, [r/o], que indica que el valor de la propiedad es de "solo lectura" (solo se puede leer y no se puede modificar)
- Una breve descripción de la propiedad

Si un objeto tiene forma plural, puede referirse a más de un objeto.

Si un objeto presenta una lista de elementos, contiene otros objetos. El nombre del elemento, que puede utilizarse en el script, aparece en negrita, mientras que el texto normal describe el modo de referirse a dicho elemento.

Los objetos pueden tener propiedades. Los nombres de las propiedades aparecen en negrita. El texto normal describe el tipo de valor que contiene la propiedad, como por ejemplo una frase. Las propiedades pueden ser un valor constante o una lista de constantes. Un script puede adoptar el valor de una propiedad y ajustarlo, a menos que detrás de la sintaxis de esa propiedad aparezca "[r/o]" ("read-only", solo lectura).

**TEMAS RELACIONADOS**
Definiciones de comando
Definiciones de objeto
Comandos de AppleScript
Usar diccionarios de AppleScript

**ABRIR**
Guía AppleScript Language Guide: referencia de clase

---



## Definiciones de comando

Los comandos son palabras de AppleScript que realizan acciones. En un diccionario AppleScript, los comandos figuran en la columna central y se designan mediante una letra "C" de color azul. La definición de un comando incluye una breve descripción de la acción que realiza el comando y de su sintaxis. La sintaxis es la manera en la que se escribe un comando en un script, y puede presentar varias partes.

Las palabras en negrita son los términos que se utilizan en un comando. Estas palabras deben escribirse tal como aparecen en el diccionario.

Las palabras en azul muestran información proporcionada por usted (por ejemplo, la identificación de un objeto o una variable definidos anteriormente).

A veces, los comandos tienen partes opcionales, que aparecen entre paréntesis. Pero no es obligatorio utilizarlas siempre.

**TEMAS RELACIONADOS**

Definiciones de clase
Usar diccionarios de AppleScript
Definiciones de evento
Definiciones de objeto

**ABRIR**

Guía AppleScript Language Guide: referencia de comandos

---



## Definiciones de evento

Una definición de un evento describe un verbo o comando empleado para aplicar una acción a un objeto que acepte scripts. Los eventos son las acciones o verbos que los objetos pueden cumplir o efectuar.

Normalmente, una definición de un evento empieza por un término y continúa con una breve descripción del término en la misma línea.

A continuación figura una lista de parámetros utilizados en la ejecución del comando. Los parámetros opcionales están entre paréntesis, y los obligatorios no.

Cada una de las entradas de un parámetro comienza con el nombre del parámetro, continúa con una descripción del formato de datos empleado con el parámetro y termina con una breve descripción del parámetro.

La definición de un comando incluye una breve descripción (el texto subrayado) de la acción que realiza el comando y de su sintaxis. La sintaxis es la manera en la que se escribe un comando en un script, y puede presentar varias partes.

Las palabras en negrita son los términos que se utilizan en un comando. Estas palabras deben escribirse tal como aparecen en el diccionario.

Las palabras en texto normal ("plain text") muestran información proporcionada por usted (por ejemplo, la identificación de un objeto o una variable definidos anteriormente).

A veces, los comandos tienen partes opcionales, que aparecen entre paréntesis. Pero no es obligatorio utilizarlas siempre.

Los comandos actúan sobre objetos. La definición de un objeto muestra la sintaxis para identificarlo, incluyendo su nombre y una breve descripción (el texto subrayado). La clase de un objeto determina ciertas características de él, como por ejemplo los comandos que pueden utilizarse con dicho objeto.

**TEMAS RELACIONADOS**

Eventos de Base de datos
Definiciones de clase
Definiciones de comando
Definiciones de objeto
Funcionamiento de AppleScript

---



## Definiciones de objeto

Los comandos actúan sobre objetos. La definición de un objeto en un diccionario de AppleScript muestra la sintaxis para identificar al objeto, incluyendo su nombre y una breve descripción (el texto subrayado). La clase de un objeto determina ciertas características de él, como por ejemplo los comandos que pueden utilizarse con dicho objeto.

Si un objeto tiene forma plural, puede referirse a más de un objeto.

Si un objeto presenta una lista de elementos, contiene otros objetos. El nombre del elemento, que puede utilizarse en el script, aparece en negrita, mientras que el texto normal describe el modo de referirse a dicho elemento.

Los objetos pueden tener propiedades. cuyos nombres aparecen en azul y negrita. El texto normal de color azul describe el tipo de valor que contiene la propiedad, como por ejemplo una frase. Las propiedades pueden ser un valor constante o una lista de constantes. Un script puede adoptar el valor de una propiedad y ajustarlo, a menos que detrás de la sintaxis de esa propiedad aparezca "[r/o]" ("read-only", solo lectura).

**TEMAS RELACIONADOS**

Utilizar AppleScript para enviar comandos a aplicaciones
Definiciones de comando
Definiciones de clase
Decir a un objeto qué tiene que hacer
Usar diccionarios de AppleScript

**ABRIR**

Guía AppleScript Language Guide: referencia de clase
Guía AppleScript Language Guide: Propiedades

---

**Comandos básicos**

## Sentencias “if”

Una sentencia "if" es un comando que se ejecuta solo si se cumplen determinadas condiciones. Cada sentencia "if" contiene una expresión booleana que se evalúa como verdadera ("true") o falsa ("false"). Los comandos incluidos en la sentencia "if" solo se ejecutan si el valor de la expresión booleana es verdadero. Una sentencia "if" termina con un "end if". Por ejemplo:

```
if amountReceived > 100 then
        set memberStatus to "special"
end if
```

En la sentencia anterior, si la expresión booleana es verdadera (la cantidad recibida es superior a 100), entonces los comandos de la sentencia "if" se ejecutarán. Si la expresión booleana es falsa, los comandos no se ejecutarán.

La sentencia "if" también puede construirse con una sentencia "else". Esta combinación se conoce como una sentencia "if-then-else". Por ejemplo:

```
if amountReceived > 100 then
        set memberStatus to "special"
else
        set memberStatus to "general"
end if
```

En la sentencia anterior, si la primera expresión booleana es verdadera (la cantidad recibida es superior a 100), entonces se ejecutarán los comandos de la primera parte de la sentencia. Si la primera expresión booleana es falsa, se ejecutarán

los comandos de la segunda parte de la sentencia (situados después de "else").

Es posible anidar sentencias una dentro de otra para controlar varias condiciones.

Para obtener más información acerca de las sentencias "if" de AppleScript, consulte la guía AppleScript Language Guide.

**TEMAS RELACIONADOS**

Repetir sentencias
Definiciones de comando
Definir variables en AppleScript
Valores constantes

**ABRIR**

Guía AppleScript Language Guide: sentencias "if"

---



# Repetir sentencias

Puede utilizar las sentencias “repeat” para realizar tareas varias veces en función de las condiciones definidas. Una sentencia "repeat" continúa ejecutando un grupo de sentencias en bucle hasta que se cumpla una condición y se abandone el bucle. Es posible anidar sentencias “repeat” una dentro de otra para responder a diferentes condiciones.

AppleScript le ofrece un abanico de sentencias “repeat” que pueden responder a diferentes condiciones.

### Sentencia “repeat (forever)”

Se repite constantemente, en un bucle. Utilice una sentencia de salida ("exit") para abandonar el bucle.

```
repeat
        -- statement
        -- statement
        if (x = 5) then exit repeat
end repeat
```

### Sentencia “repeat (número) times”

Se repite el número de veces especificado.

```
repeat 5 times
        -- statement
        -- statement
end repeat
```

### Sentencia “repeat while”

Se repite mientras una condición especificada sea verdadera.

```
set x to 1
repeat while (x < 5)
        -- statement
```

```
        -- statement
        set x to (x + 1)
end repeat
```

---

**Sentencia “repeat until”**

Se repite hasta que una condición especificada sea verdadera.

```
set x to 1
repeat until (x = 5)
        -- statement
        -- statement
        set x to (x + 1)
end repeat
```

---

**Sentencia “repeat with (variable) from (valor_inicial) to (valor_final)”**

Asigna a la variable el "valor_inicial" y, a continuación, incrementa la variable cada vez que se completa el bucle. La repetición finaliza cuando la variable alcanza el "valor_final".

```
repeat with x from (5) to (25)
        -- statement
        -- statement
end repeat
```

---

**Sentencia “repeat with (variable) in (lista)”**

Asigna a la variable el valor del primer elemento de la lista y, cada vez que se completa el bucle, asigna a la variable el valor del elemento siguiente. La repetición finaliza cuando se han utilizado todos los elementos de la lista.

```
set myList to {234, 342, 681, 874}
repeat with x in myList
        -- statement
        -- statement
end repeat
```

---

Para obtener más información acerca de las sentencias "repeat" de AppleScript, consulte la guía AppleScript Language Guide.

**TEMAS RELACIONADOS**

Sentencias “if”
Definir variables en AppleScript
Valores constantes

**ABRIR**

Guía AppleScript Language Guide: sentencias "repeat"

## Valores constantes

Una constante es una palabra reservada de AppleScript que contiene un valor predeterminado. AppleScript contiene numerosas contantes que puede usar para obtener información acerca de objetos, realizar comparaciones y efectuar cálculos aritméticos. Las aplicaciones y otros objetos pueden definir sus propias constantes. Dichas constantes se definen en las definiciones de clase del valor.

Para ver el valor de una constante, pegue el siguiente ejemplo en Editor AppleScript y, a continuación, haga clic en Ejecutar. Haga clic en Resultado para ver el valor de la constante.

El resultado es también una constante. Se trata de una constante de script que se refiere al valor generado por un comando.

```
get the current date
```

Algunos otros ejemplos de constantes:

- Constantes aritméticas: pi.
- Constantes booleanas: true, false.
- Constantes de fecha y hora: minutes, hours, days, weeks.
- Constantes de texto: return, space, tab.
- Constantes de script: anything, current application, it, me, missing value, my, result.

Para obtener más información acerca de las constantes de AppleScript, consulte la guía AppleScript Language Guide.

**TEMAS RELACIONADOS**

Definir variables en AppleScript
Definiciones de objeto
Definiciones de clase
Definiciones de evento

**ABRIR**

Guía AppleScript Language Guide: constantes

---

**Comandos básicos**

## Definir variables en AppleScript

Una variable es un contenedor creado dentro de un script que sirve para guardar un valor. Al crear una variable, se le asigna un nombre, también denominado identificador. Al ejecutar un script, puede asignar un valor a la variable, acceder al valor o modificarlo. Las variables solo duran mientras el script se ejecuta. Cuando el script se cierra, el valor de la variable se pierde.

- En estos ejemplos se muestra cómo especificar un identificador exclusivo y asignarle un valor.

```
set myVar to "153"

set x to "Quarterly Reports"
```

```
set x to false
```

Para ver cómo se almacenan los valores de una variable, pegue estos ejemplos en Editor AppleScript y, a continuación, haga clic en Ejecutar. Haga clic en Resultado para ver el valor guardado en la variable.

Para obtener más información acerca de las variables de AppleScript, consulte la guía AppleScript Language Guide.

**TEMAS RELACIONADOS**

Valores constantes
Sentencias "if"
Repetir sentencias

**ABRIR**

Guía AppleScript Language Guide: variables

---

**Comandos básicos**

## Especificar la ubicación de una aplicación

Las aplicaciones de OS X se entregan en forma de "paquetes". Un paquete es un directorio especial que contiene las carpetas y los archivos de los que consta una aplicación. Por este motivo, los comandos "path to" e "info for" definidos en las adiciones estándar pueden dar resultados diferentes para una aplicación.

Si utiliza el comando "path to" para una aplicación, la ruta normalmente termina en ".app". Es posible que en el Finder no se muestren las extensiones de los archivos como parte de su nombre, pero aún así puede utilizarlas para identificar un archivo en un script.

Si utiliza el comando "info for" para obtener información acerca de una aplicación, la propiedad "Folder" del resultado suele ser "true". Puede obtener más información acerca del archivo y su nombre si utiliza propiedades adicionales del resultado, como "extension hidden", "name extension" o "displayed name".

**TEMAS RELACIONADOS**

Comando "path to"
Comando "info for"
Referirse a carpetas, archivos y discos
Comando "choose application"
Comandos de creación de scripts para el Finder

---

**Comandos para el Finder**

## Referirse a carpetas, archivos y discos

AppleScript le permite referirse a ciertas carpetas, archivos y discos por su nombre, sin necesidad de incluir una ruta. AppleScript puede identificar las ubicaciones de estos elementos, tales como la carpeta Documentos, el escritorio, la carpeta de inicio y el disco de arranque.

---

- En este ejemplo se muestra cómo referirse a una carpeta específica incluyendo su nombre en un script.

```
set x to the path to the desktop folder
```

**TEMAS RELACIONADOS**

Referirse al Finder en los scripts
Comando “path to”
Comando “info for”
Comandos de creación de scripts para el Finder

Comandos para el Finder

## Referirse al Finder en los scripts

Para usar comandos del Finder, debe enviar los comandos al Finder usando una sentencia "tell" tal como lo haría para enviar comandos a una aplicación.

- A continuación se proporciona un ejemplo de uso de la sentencia “tell” en un script. La sentencia “tell” dirige la función de número de ventanas (number of windows) a “Finder.app”.

```
tell application "Finder.app"
        return the number of windows
end tell
```

**TEMAS RELACIONADOS**

Clase “Finder”
Decir a un objeto qué tiene que hacer
Referirse a carpetas, archivos y discos
Definiciones de objeto
Comandos de creación de scripts para el Finder

**ABRIR**

Guía AppleScript Language Guide: especificadores de objeto

Comandos para el Finder

## Comandos de creación de scripts para el Finder

A continuación se incluye una lista con varios comandos que puede utilizar en el Finder:

Comando “activate”

Clase “alias file”

Clase “alias list”

Clase “application file”

Comando “clean up”

Comando “close”

Clase “column”

Clase “computer-object”

Clase “container”

Comando “copy”

Comando “count”

Comando “data size”

Comando “delete”

Clase “desktop window”

Comando “desktop”

Clase “disk”

Clase “document file”

Comando “duplicate”

Comando “eject”

Comando “empty trash”

Comando “exists”

Clase “file”

Clase “file information”

Comando “find”

Clase “Finder”

Clase “Finder window”

Clase “folder”

Comando “get”

Clase “icon view options”

Clase “information window”

Clase “Internet location file”

Clase “item”

Clase “list view options”

Comando “make”

Comando “move”

Comando “moving folder window for”

Comando “open”

Clase “package”

Comando “path to”

Clase “preferences”

Clase “preferences window”

Comando “print”

Comando “quit”

Referirse a carpetas, archivos y discos

Referirse al Finder en los scripts

Comando “reveal”

Comando “select”

Comando “set”

Comando “sort”

Propiedad “trash”

Comando “update”

Clase “window”

---

**Comandos para el Finder**

## Comandos estándar de adición de scripting

La adición de scripting proporciona muchos comandos útiles que se pueden incluir en los scripts.

Comando “adding folder items to”

Comando “beep”

Comando “choose application”

Comando “choose color”

Comando “choose file”

Comando “choose file name”

Comando “choose folder”

Comando “choose from list”

Comando “choose URL”

Comando “clipboard info”

Comando “close access”

Comando “closing folder window for”

Comando “current date”

Comando “delay”

Clase “dialog reply”

Clase “display”

Comando “display dialog”

Comando “do shell script”

Comando “download”

Clase “FTP item”

Comando “get eof”

Comando “info for”

Comando “list disks”

Comando “list folder”

Comando “load script”

Comando “mount volume”

Comando “offset”

Comando “open for access”

Comando “open location”

Comando “opening folder”

Comando “path to”

Clase “POSIX path”

Comando “random number”

Comando “read”

Comando “round”

Comando “run script”

Comando “say”

Comando “scripting components”

Comando “set eof”

Comando “set the clipboard to”

Comando “set volume”

Comando “store script”

Comando “summarize”

Comando “system attribute”

Comando “time to GMT”

Comando “upload”

Clase “URL”

Comando “write”